A Principled Study of the Design Tradeoffs for Autonomous Trading Agents

Ioannis A. Vetsikas*
Computer Science Dept., Cornell University
Ithaca, NY 14853, USA
vetsikas@cs.cornell.edu

Bart Selman
Computer Science Dept., Cornell University
Ithaca, NY 14853, USA
selman@cs.cornell.edu

ABSTRACT

In this paper we present a methodology for deciding the bidding strategy of agents participating in a significant number of simultaneous auctions, when finding an analytical solution is not possible. We decompose the problem into sub-problems and then use rigorous experimentation to determine the best partial strategies. In order to accomplish this we use a modular, adaptive and robust agent architecture combining principled methods and empirical knowledge. We applied this methodology when creating WhiteBear, the agent that achieved the highest score at the 2002 International Trading Agent Competition (TAC). TAC was designed as a realistic complex test-bed for designing agents trading in e-marketplaces. The agent faced several technical challenges. Deciding the optimal quantities to buy and sell, the desired prices and the time of bid placement was only part of its design. Other important issues that we resolved were balancing the aggressiveness of the agent's bids against the cost of obtaining increased flexibility and the integration of domain specific knowledge with general agent design techniques. We present our observations in dealing with these design tradeoffs and back up our conclusions with empirical results.

Keywords

agent-mediated electronic commerce, bidding agents, determining bidding strategies, electronic marketplaces, simultaneous auctions

1. INTRODUCTION

Auctions are becoming an increasingly popular method for transacting business either over the Internet (e.g. eBay) or even between businesses and their suppliers. While a good deal of research on auction theory exists, this is mostly from the point of view of auction mechanisms (for a survey see [9]). Strategies for bidding in an auction for a single item are also known. However, in practice agents (or humans) are rarely interested in a single item. They wish to bid in several auctions in parallel for multiple interacting goods. In this case they must bid more intelligently in order to get the exactly what they need. For example a person may wish to buy a TV and a VCR, but if she does not have a "flexible plan", she may only end up acquiring the VCR. Goods are called complementary if the value of acquiring both together is higher than the sum of their individual values. On the other hand if that person bids for VCRs in several auctions, she may end up with more than one. Goods are called substitutable if the value of acquiring two of them is less than the sum of their individual values. There have been relatively few studies about agents bidding in multiple simultaneous auctions and they mostly involve bidding for substitutable goods (e.g. [1], [10], [8]). In this and other related work, such as [7], researchers tested their ideas about agent design in smaller market games that they designed themselves. Time was spent on the design and implementation of the market, but there was no common market scenario that researchers could focus on and use to compare strategies. In order to provide a universal test-bed, Wellman and his team [16] designed and organized the Trading Agent Competition (TAC). It is a challenging benchmark domain which incorporates several elements found in real marketplaces in the realistic setup of travel agents that organize trips for their clients. It includes several complementary and substitutable goods traded in a variety of different auctions. However, instead of bidding for bundles of goods and letting the auctioneer determine the final allocation that maximizes its income (as in the combinatorial auction mechanism, see [4], [11]), in this setting the computational cost is shifted to the agents who have to deal themselves with the complementarities and substitutabilities between the goods.

In order to tackle this problem, we investigate a methodology for decomposing the problem into several subparts and use systematic experimentation to determine the strategies that work best for the problem in question. We chose TAC as our test-bed and we implemented our methodology in WhiteBear, the agent that achieved the top score (1st place) in this year's (2002) TAC. An earlier version of Whitebear, which was not tuned using the methodology decribed in this paper, reached 3rd place in last year's competition. Our goal is to provide a scalable and robust bidding agent that incorporates principled methods and empirical knowledge. As part of our experimentation, we studied design tradeoffs applicable to many market domains. We examine the tradeoff of the agent paying a premium either for acquiring information or for having a wider range of actions available against the increased flexibility that this information provides. We also examined how agents of varying degrees of bidding aggressiveness perform in a range of environments against other agents of different aggressiveness. We find that there is a certain optimal level of "aggressiveness": an agent that is just aggressive enough to implement its plan outperforms agents who are either not aggressive enough or too aggressive. We will show that the resulting agent strategy is quite robust and performs well in a range of agent environments.

*The primary author is a student. This person is also the corresponding author.

We also show that even though generating a good plan is crucial for the agent to maximize its utility, it may not be necessary to compute the optimal plan. In particular, we will see that a randomized greedy search methods produces plans that are close to optimal and of sufficient quality to allow for effective bidding. The use of such a randomized greedy strategy results in a more scalable agent design. Overall our agent is adaptive and robust. Moreover, it appears that its design elements are general enough to work well under a variety of settings that requires bidding in a large number of multiple simultaneous auctions.

The paper is organized as follows. In the section 2, we give the definition of the general problem our methodology is applied and the rules of the TAC market game. In section 3, we present our methodology and how it is applied to the TAC domain. In section 4 we explain in detail the controlled experiments needed in order to implement our methodology and to study the agent design tradeoffs. In section 5 we present the results and our observations from the TAC competition. Finally in section 6 we discuss possible directions for future work and conclude.

2. TRADING GOODS IN SIMULTANEOUS AUCTIONS

We first present the problem setting in section 2.1. In section 2.2, we present the description of the TAC game together with the reasons why this game captures many of the issues of the general problem setting.

2.1 General Problem Setting

The general problem setting that we deal with involves several autonomous agents, which wish to trade commodities in order to acquire the goods that they need. There is a predefined time window during which the trades can take place (defining the duration of each "game"), after which each agent calculates the payoff to itself. The agents are not allowed to cooperate in any explicit way (even though implicit cooperation might arise from their behavior) and they are also assumed to be self interested. In particular, each agent i is trying to maximize its own utility function

$$U_i(\theta_i, C_i, t_i)$$

where θ_i is type of the agent (parameters selected randomly from a given distribution, that influence the utility function), C_i is the set of commodities that the agent owns and t_i is the net monetary transfer, that is the algebraic sum of payments for selling goods minus the cost of buying goods. In most cases we can assume that the utility is linear in the monetary transfers t_i, namely that $U_i(\theta_i, C_i, t_i) = u_i(\theta_i, C_i) + t_i$. The combination of goods C_i owned should include several complementary and substitutable goods in order for the game to be interesting; otherwise one might be able to find an equilibrium to the game analytically.

The mechanism used in order to exchange commodities are several different auctions during which each unit of a certain commodity is traded in exchange for a monetary payment. We will assume that there is *no discriminatory pricing* in these auctions, which means that if two agents wish to buy the same good at the same time they will have make the same payment. We will also assume that *similar goods are sold in auctions with similar rules*[1]. Other than that we allow the auctions to have a wide variety of rules, the most important of which are:

1. Agents may act as *buyers only*, *sellers only* or *both* in the auctions, so we can have *single-sided* and *double-sided auctions*. In case e.g. they act as buyers then an external source would have to provide (input) goods into the system and remove money from it.
2. There can be a *limited* or an *infinite number* of units for each commodity. The lowest number of units is 1 in the case we have a limited number.
3. Auctions can clear *continuously*, *several times* or *only once*. The first case means that trades can take place at any time, while the last that trades take place exactly once, when the auction closes.
4. Auctions can close *at set known times*, or at *unspecified times* (e.g. determined by random parameters).
5. Clearing prices can be determined *only by the bids of the agents* or *by external parameters* as well (e.g. the price set by an external seller). Pricing in the first case in particular can follow any pricing scheme (e.g. N^{th}, $(N+1)^{th}$ highest bid etc. where N is the number of identical goods for sale in the auction).

[1]We make this assumption because this usually makes reasoning about bidding strategies easier. However there are several cases in which our methodology would work even if this assumption does not hold for all auctions.

The question we are interested in answering is what bids to place at each auction. There are therefore 3 main parameters to determine: the *quantity* of each good one wishes to buy, the *prices* offered for each individual unit and the *times* at which the bids are placed.

2.2 TAC: A description of the Market Game

The TAC game encapsulates most of the issues of the general problem and is thus an appropriate test-bed for evaluating our agent design.

- Each auction has rules which cover the various options discussed in the previous section: some auctions are *single-sided* and others *double-sided*, some offer a *limited* and some an *unlimited* number of identical goods, some clear *continuously* and others *only once*, some close at *preset times* and at *random times*. Also some auction clear at *prices determined by the agents' bids*, others at *prices determined by outside sellers*.
- There are 28 auctions running in parallel (and in fact our strategies and methodology scales well and would also work for a larger TAC game with many more auctions). This setting is too complex to allow for analytical derivation of equilibrium (or optimal) strategies.
- A number of different tradeoffs are present in this game, which makes the determination of an appropriate bidding strategy a difficult design problem. For a detailed analysis of these tradeoffs, see section 3.2.
- The TAC setting is designed to model a realistic market place setting, as might be encountered by, for example, a travel agent. It also includes several complementary and substitutable goods and a complex utility function.

In the Trading Agent Competition, an agent competes against 7 other agents in each game. Each agent is a travel agent with the goal of arranging a trip to Tampa for $CUST$ customers. To do so it must purchase plane tickets, hotel rooms and entertainment tickets. Each good is traded in separate simultaneous online auctions. The agents send their bids to the central server and are informed about price quotes and transaction information. Each game lasts for 12 minutes (720 seconds). For the full details of the mechanism and the rules of TAC, see the url: http://www.sics.se/tac.

Plane tickets (8 auctions): There is only one flight per day each way with an infinite amount of tickets, which are sold in separate, continuously clearing auctions in which prices follow a random walk. For each flight auction a hidden parameter x is chosen. The prices tend to increase every 30 seconds or so and the hidden parameter influences the change. The agents may not resell tickets.

Hotels (8 auctions): There are only two hotels in Tampa: the Tampa Towers (cleaner and more convenient) and the Shoreline Shanties (the not so good and expected to be the less expensive hotel). There are 16 rooms available each night at each hotel. Rooms for each of the days are sold by each hotel in separate, ascending, open-cry, multi-unit, 16^{th}-price auctions. A customer must stay in the same hotel for the duration of her stay and placed bids may not be removed. One randomly selected auction closes at minutes 4 to 11 (one each minute, on the minute). No prior knowledge of the closing order exists and agents may not resell rooms they have bought.

Entertainment tickets (12 auctions): They are traded (bought and sold) among the agents in continuous double auctions (stock market type auctions) that close when the game ends. Bids match continuously. Each agent starts with an endowment of 12 random tickets and these are the only tickets available in the game.

The *type of each agent* is determined by the preferences of its clients. Each customer i has a preferred arrival date PR_i^{arr} and a preferred departure date PR_i^{dep}. She also has a preference for staying at the good hotel represented by a utility bonus UH_i as well as individual preferences for each entertainment event j represented by utility bonuses $UENT_{i,j}$.

The parameters of customer i's itinerary that an agent has to decide upon are the assigned arrival and departure dates, AA_i and AD_i respectively, whether the customer is placed in the good hotel GH_i (which takes value 1 if she is placed in the Towers and 0 otherwise) and $ENT_{i,j}$ which is the day that a ticket of the event j is assigned to customer i (this is e.g. 0 if no such ticket is assigned). Let $DAYS$ be the total number of days and ET the number of different entertainment types.

The utility that the travel plan has for each customer i is:

$$\begin{aligned} util_i = & 1000 + UH_i \cdot GH_i \\ & + \sum_{d=AA_i}^{AD_i} \max_j \{ UENT_{i,j} \cdot \mathbf{I}(ENT_{i,j} = d) \} \\ & - 100 \cdot \left(|PR_i^{arr} - AA_i| + |PR_i^{dep} - AD_i| \right) \end{aligned} \quad (1)$$

if $1 \leq AA_i < AD_i \leq DAYS$,
else $util_i = 0$, because the plan is not feasible.

It should be noted that only one entertainment ticket can be assigned each day and this is modeled by taking the maximum utility from each entertainment type on each day. We assume that an unfeasible plan means no plan (e.g. $AA_i = AD_i = 0$). The function $\mathbf{I}(bool\_expr)$ is 1 if the $bool\_expr$ =TRUE and 0 otherwise.

The total income for an agent is equal to the sum of its clients' utilities. Each agents searches for a set of itineraries (represented by the parameters AA_i, AD_i, GH_i and $ENT_{i,j}$) that maximize this profit while minimizing its expenses.

3. OUR PROPOSED METHODOLOGY AND THE AGENT ARCHITECTURE

In the first year that the TAC was organized (2000), the auctions rules did not introduce any real tradeoffs in the design of the agent. A dominant bidding strategy was found by some of the top scoring agents: *buy everything at the last moment and bid high (the marginal utility) for hotel rooms* [5]. This happened because the hotel auctions were not closing at random intervals and the prices of plane tickets remained approximately the same over time. Therefore most top-scoring teams concentrated on solving the optimization problem of maximizing the utility, since bid prices and bid placement times were not an issue. The rule changes in the 2001 TAC introduced the tradeoffs that made the game interesting. Most teams decided to use a learning strategy (e.g. ATTac, the winner of the 2000 TAC, used a boosting-based method [13]) in order to predict the prices at different times and also decided that decomposing the problem completely was probably not to their best interest as there are obvious dependencies among the quantity, the price and the placement time of the each bid. Given our research interests involve bidding strategies for agents and experimentation on how the different agent behaviors influence the behavior of the multi-agent system, we decided to investigate in the opposite direction and further decompose the problem.

The high-level description of the methodology we propose is:

A. Decompose the problem into subproblems:
  1. Decide the quantities to buy assuming that everything will be bought at current prices (*optimize utility*)
  2. For each different auction type (and good) do:
    a. Determine boundary "partial strategies" for this auction
    b. Generate "intermediate" strategies. Main approaches:
      - combine the boundary strategies, or
      - modify them using empirical knowledge from the domain

B. Use rigorous experimentation to evaluate partial strategies:
  1. Keep other partial strategies fixed if possible
  2. Experiment with different mixtures of agents as follows:
    a. Keep fixed the agents using intermediate strategies
    b. Vary the number of agents using boundary strategies
  3. Evaluate differences in performance using statistical tests
  4. Determine best strategy overall (in all possible mixtures)

The first part of our methodology requires little further explanation; for each different auction type, which corresponds to a different commodity type, we compute (usually analytically but sometimes also using domain knowledge) the boundary "partial strategies" that are possible.[2] We then combine parts of the boundary strategies or modify some of their parts to form intermediate strategies that behave between the extreme bounds (e.g. if the one boundary strategy will place a bid at price p_{low} and the other at price p_{high} in a certain case, then the intermediate strategy should place its bid at price $p : p_{low} \leq p \leq p_{high}$). For the specifics of how this part of the methodology is applied to the TAC domain, see section 3.2. The quantities placed in each bid are determined independently by maximizing the utility of the agent assuming that all the goods are bought at some predicted prices and that every unit will be bought instantly. A dedicated module called "the planner" is doing this task. The planer for the TAC problem is described in section 3.1.

The second part of our methodology deals with the way experiments are run in order to determine the best combination of partial strategies. Each set of experiments is designed to evaluate the partial strategies in different mixes of agents. Determining the mixture of agents is an issue of paramount importance in multi-agent systems, since the performance of each strategy depends on the competition offered by the other agents. In order to explore the whole spectrum of mixtures we propose to keep a fixed number of agents who are using the intermediate strategies, while systematically changing the mixture of agents using the boundary cases. For example we start with all "boundary strategy" agents using the low-bidding strategy and in each subsequent experiment replace some with ones the high-bidding strategy until in the last experiment we have only the latter type. This will explore sufficiently the different multi-agent environment that the agents can participate in, since the behavior caused by the intermediate strategies is within the bounds of the behavior caused by the boundary strategies. For more details about how this helps us determine the experiments to run in the case of TAC in order to choose the strategy that performs the best across all agent mixtures, see section 4. Using this methodology also allows us to derive general observations about the behavior of certain strategies on different domains.

[2]We called these strategies partial because they only deal with one particular type of auctions.

This methodology and the desire to have a scalable and general system imposes some requirements on the agent architecture that we must use. For an agent architecture to be useful in a general setting, it must be *adaptive, flexible* and *easily modifiable*, so that it is possible to make changes on the fly and adapt the agent to the system in which it is operating. In addition, as information is gained by participation in the system, the agent architecture must allow and *facilitate the incorporation of the knowledge obtained*. The architecture should support interchangeable parts so that different strategies are easy to implement and change, otherwise running experiments with agents using the different strategies would be quite time consuming and also the incorporation of domain specific knowledge would be an arduous task. These were lessons that we incorporated in to the design of our architecture.

The general architecture that we used follows the "Sense Model Plan Act (SMPA)" architecture (this name originated with Brooks [2]). Other trading agents, such as, e.g., [6], have used a similar global design. Including the decomposition in the bidding section of the architecture that we introduced, the overall architecture can be summarized as follows:

while (not end of game) {
1. Get price quotes and transaction information
2. Calculate price estimates
3. *Planner*: Form and solve optimization problem
4. *Bidder*: Bid to implement plan
   Determine each bid independently of all other bids
   Use a different "partial strategy" for each different bid }

This architecture is highly modular and each component of the agent can be replaced by another or modified. In fact parts of the components themselves are also substitutable (e.g. the partial strategies). One last requirement that is desired is to design our the modules of the agent to be as fast and adaptive as possible without sacrificing efficiency. Speed is not so much of a problem in the TAC game, since each agent can spend up to 30-60 seconds deciding its next bids, but in other domains it is crucial to react fast (within seconds) to domain information and other agents' actions.

In the next sections, we present how the first part of our methodology is applied in the TAC domain, that is the decomposition into subproblems. In section 3.1 we present the planner module and in section 3.2 the selection of boundary and intermediate strategies.

3.1 Planner

The planner is a module of our architecture. In order to formulate the optimization problem that it solves, it is necessary to estimate the prices at which commodities are expected to be bought or sold. We started from the "priceline" idea presented in [6] and we simplified and extended it where appropriate. We implemented a module which calculates *price estimate vectors* (PEV). These contain the value (price) of the x^{th} unit for each commodity. Let $PEV_d^{arr}(x)$, $PEV_d^{dep}(x)$, $PEV_d^{goodh}(x)$, $PEV_d^{badh}(x)$ and $PEV_{d,t}^{ent}(x)$ be the PEVs. For some goods this price is the same for all units, but for others it is not; e.g. buying more hotel rooms usually increases the price one has to pay, since there is a limited supply. Other information to account for is the fact that some commodities once bought cannot be sold, so in that case they have to be considered as "sunk cost", so their PEV is 0. For some goods these values are known accurately and for some others they are estimated based on the current ask and bid prices.

Let us define the operator $\sigma(f(x), z) = \sum_{i=1}^{z} f(i)$. The utility function that the agent wishes to maximize is:

$$\max_{AA_c, AD_c, GH_c, ENT_{c,t}} \Big\{ \sum_{c=1}^{CUST} util_c - COST \Big\} \tag{2}$$

where the cost of buying the resources is

$$\begin{aligned} COST = \textstyle\sum_{d=1}^{DAYS} \big[&\sigma\big(PEV_d^{arr}(x), \textstyle\sum_{c=1}^{CUST} \mathbf{I}(AA_c = d)\big) \\ &+ \sigma\big(PEV_d^{dep}(x), \textstyle\sum_{c=1}^{CUST} \mathbf{I}(AD_c = d)\big) \\ &+ \sigma\big(PEV_d^{goodh}(x), \textstyle\sum_{c=1}^{CUST} [GH_c \cdot \mathbf{I}(AA_c \le d < AD_c)]\big) \\ &+ \sigma\big(PEV_d^{badh}(x) \textstyle\sum_{c=1}^{CUST} [(1 - GH_c) \cdot \mathbf{I}(AA_c \le d < AD_c)]\big) \\ &+ \textstyle\sum_{t=1}^{ET} \sigma\big(PEV_{d,t}^{ent}(x), \textstyle\sum_{c=1}^{CUST} \mathbf{I}(ENT_{c,t} = d)\big) \big] \end{aligned} \tag{3}$$

Once the problem has been formulated, the planner must solve it. This problem is NP-complete, but for the size of the TAC problem an optimal solution, that is the type and total quantity of commodities that should be traded to achieve maximum utility, can usually be produced fast. However in order to create a more general algorithm we realized that it should scale well with the size of the problem and should not include elaborate heuristics applicable only to the TAC problem. Thus we chose to implement a greedy algorithm: the order of customers is randomized and then each customer's utility is optimized separately. This is done a few hundred times in order to maximize the chances that the solution will be optimal most of the time. In practice we have found the following additions (that were not reported by anyone else) to be quite useful:

1. Compute the utility of the plan $\mathcal{P}_1$ from the previous loop before considering other plans. Thus the algorithm always finds a plan $\mathcal{P}_2$ that is at least as good as $\mathcal{P}_1$ and there are relatively few radical changes in plans between loops. We observed empirically that this prevented radical bid changes and improved efficiency.
2. We added a constraint based on the idea of strategic demand reduction [15], that dispersed the bids of the agent for resources in limited quantities (hotel rooms in TAC). Plans, which demanded many hotel rooms for any single day, were not considered. This leads to some utility loss in rare cases. However, bidding heavily for one room type means that overall demand will very likely be high and therefore prices will skyrocket, which in turn will lower the agent's score significantly. We observed empirically that the utility loss from not obtaining the best plan tends to be quite small compared to the expected utility loss from rising prices.

We have also verified that this randomized greedy algorithm gives solutions which are often optimal and never far from optimal. We checked the plans (at the game's end) that were produced by 100 randomly select runs and observed that over half of the plans were optimal and on average the utility loss was about 15 points (out of 9800 to 10000 usually[3]), namely close to 0.15%. Compared to the usual utility of 2000 to 3000 that our agents score in most games, they achieved about 99.3% to 99.5% of optimal. These observations are consistent with the ones about the related greedy strategy in [12] from which the initial idea for this algorithm was taken. Considering that at the beginning of the game the optimization problem is based on inaccurate values, since the closing hotel prices are not known, an 100%-optimal solution is not necessary and can be replaced by our near-optimal approximation. As commodities are bought and the prices approach their closing values, most of the commodities needed are already bought and we have observed empirically that bidding is rarely affected by the generation of near-optimal solutions instead of optimal ones.

This algorithm takes approximately one second to run through 500 different randomized orders and compute an allocation for each. Our test bed was a cluster of 8 Pentium III 550 MHz CPU's, with each agent using no more than one cpu. This system was used for

[3] These were the scores of the allocation at the end of the game (no expenses were considered).

all our experiments and our participation in the TAC.[4] In summary, our planner is fast, relatively domain-independent, and performs near-optimally. Moreover, using a close to optimal but nevertheless non-optimal plan does not effect the agent's overall performance.

3.2 Bidding Strategies

Once the planner has been generated the desired types and quantities of each good, the bidder module places separate bids for all these goods. According to our methodology, we find strategies for each different set of auctions and this procedure is described in this section. We use principled approaches where applicable together with empirical knowledge that we acquired during the competition. In fact every every participating team, including ours, used empirical observations from the games it participated in (some 2000 games over the 2001 and 2002 TAC) in order to improve its strategy. In the next sections we describe how the strategies are generated for the different auctions and the tradeoffs that our agent faced.

3.2.1 Bid Aggressiveness

Bidding for hotel rooms poses some interesting questions. The main issue in this case is how *aggressively each agent should bid* (the level of the prices it submits in its bids). If it bids low it might get outbid, while if it bids high it is likely to enter into price wars with the other agents.

The first boundary strategy is to place low bids: the agent bids an increment higher than the current price. The agent also bids progressively higher for each consecutive unit of a commodity for which it wants more than one unit. E.g. if the agent wants to buy 3 units of a hotel room, it might bid 210 for the first, 250 for the second and 290 for the third. This is the lowest (L) possible aggressiveness since the agent will never wish to bid less. The other boundary strategy is that the agent bids progressively closer to the marginal utility δU as time passes[5]. Since the agent will likely lose money if it bids above the marginal utility, this is the highest (H) possible aggressiveness. Now that the boundary strategies are set our methodology suggest that we try to combine these into intermediate strategies. We therefore selected the following compromise: the agent that bids like the aggressive (H) agent for rooms that have a high marginal utility δU and bids like the non-aggressiveness (L) agent otherwise. This is the agent of medium (M) aggressiveness. One further improvement, which was judged necessary for the 2002 TAC, is to use historical data to determine the price of the hotel auction which closes first and that is because we observed that our agent was getting outbid while the bids were still low.

As far as the timing of the bids is concerned, there is little ambiguity about what the optimal strategy is. The agent waits until the first hotel auction is about to close to places its first bids. The reason for this is that it does not wish to increase the prices earlier than necessary nor to give away information to the other agents. We also observed empirically that an added feature which increases performance is to place bids for a small number of rooms at the beginning of the game at a very low price (whether they are needed or not). In case these rooms are eventually bought, the agent pays only a very small price and gains increased flexibility in implementing its plan.

3.2.2 Paying for adaptability

The purchase of flight tickets presents an interesting dilemma as well. We have calculated (based on the model of price change described in the rules) that ticket prices are expected to increase approximately in proportion to the square of the time elapsed since the start of the game. This means that the more one waits the higher the prices will get and the increase is more dramatic towards the end of the game. From that point of view, if an agent knows accurately the plan that it wishes to implement, it should buy the plane tickets immediately. On the other hand, if the plan is not known accurately (which is usually the case), the agent should wait until the prices for hotel rooms have been determined. This is because buying plane tickets early restricts the flexibility (adaptability) that the agent has in forming plans: e.g. if some hotel room that the agent needs becomes too expensive, then if it has already bought the corresponding plane tickets, it must either waste these, or pay a high price to get the room. *An obvious tradeoff exists in this case, since delaying the purchase of plane tickets increases the flexibility of the agent and hence provides the potential for a higher income at the expense of some monetary penalty.*

One way to solve this is to use a cost-benefit analysis. In this case the cost of deferring purchase can be computed, but in order to estimate the benefit from delayed buying, one must use models of the opponents which are not too easy to obtain. Our first step is to decide on boundary strategies. Since the only issue is the time of bid placement, two obvious strategies are to buy everything at the beginning or to defer all the tickets purchases at a much later time. Initially we set this later time to be right after 2 (out of the 8) hotel auctions have closed. The reason for this is that at that time the intentions of the other agents can be partially observed by their effect on the auctions' bid prices and thus after this time the room prices approximate sufficiently their potential closing prices. Hence a plan generated at that time is usually quite similar to the optimal plan for known closing prices. Another reason is that since ticket prices are expected to increase approximately in proportion to the square of the time elapsed, the price increases after this point tend to be prohibitive. However this is still not a very good boundary case; a further improvement is to buy some tickets at the start of the game. We buy about 50% of the tickets at the beginning: these are the "almost certain to be used" tickets (computed based on the client preferences and the ticket prices) and we have empirically observed that these tickets are rarely wasted. Given these boundary strategies, we first obtained an intermediate strategy by modifying the latter to wait until only 1 (the first) hotel auction closes. Another intermediate strategy comes from the idea of strategic demand reduction [15]: we compute the minimum number of tickets which, if left unpurchased, will allow the agent to complete its itineraries even if it fails to buy a hotel room on days during which it wishes a lot of rooms. This strategy buys 80%-100% of the tickets at the beginning.

An improvement (for agents who defer the purchase of some tickets) was obtained by estimating the likelihood of price increases. This information is then used to bid earlier for tickets whose price is very likely to increase and to wait more for tickets whose price is expected to increase little or none. We calculated that the agent approximately halves the cost it would otherwise pay for the deferred purchases. The full details can be found in a workshop paper which described the strategy we used in the 2001 TAC [14]. A further improvement (especially for agents who buy most tickets at the beginning) was obtained by using historical averages of the hotel prices in previous games to set the PEVs at the beginning of the game, since the planner gives a much more accurate plan in this way.

3.2.3 Entertainment

The entertainment tickets do not present us with a challenging

[4]During the competition only one processor was used, but during the experimentations we used all 8, since 8 different instantiations of the agent were running at the same time.

[5]The marginal utility δU for a particular hotel room is the change in utility that occurs if the agent fails to acquire it. In fact for each customer i that needs a particular room we bid $\frac{\delta U}{\sqrt{z}}$ instead of δU, where z is the number of rooms which are still needed to complete her itinerary. We do this, based on empirical observations, in order not to drive the prices up prematurely.

Experiments		WB-N2L	WB-M2L	WB-M2M	WB-M2H
Exp 1	agent 1	2087	2238	2387	2429
	agent 2	2087	2274	2418	2399
(144)	average	2087	2256	2402	2414
Exp 2	(200)	3519	3581	3661	3656

Table 1: Average scores of agents WB-N2L, WB-M2L, WB-M2M and WB-M2H. For experiment 1 the scores of the 2 instances of each agent type are also averaged. *The number inside the parentheses is the total number of games for each experiment and this will be the case for every table.*

		WB*xSM	WB*xSH	WB*x2M	WB*x2H
#	x=M	1941	1887	1744	1677
games	x=A	1729	1645	1686	1706
(206)	Difference?	✓	✓	×	×

Table 4: The effect of using historical averages in the PEVs. Early bidding agents benefit the most from this.

tradeoff. Therefore we only used the following strategy. The agent buys (sells) the entertainment tickets that it needs (does not need) for implementing its plan at a price equal to the current price plus (minus) a small increment. The only exceptions to this rule are:
(i) At the game's start and depending on how many tickets the agent begins with, it will offer to buy tickets at low prices, in order to increase its flexibility at a small cost. Even if these tickets are not used the agent sometimes manages to sell them for a profit.
(ii) The agent will not buy (sell) at a high (low) price, even if this is beneficial to its utility, because otherwise it helps other agents. This restriction is somewhat relaxed at 11:00, in order for the agent to improve its score further, but it will still avoid some beneficial deals if these would be very profitable for another agent.

4. EXPERIMENTAL RESULTS

In this section we describe the controlled experiments we performed (the majority of which were based on agent mixtures designated by our methodology) in order to determine the "best overall strategy" and the conclusions we drew from them conserving the tradeoff described in section 3.2. To distinguish between the different strategies (or if you prefer versions of the agent), we use the notations WB-xyz and WB*xyz,[6] where (i) x is M if the agent models the plane ticket prices, N if this feature is not used and A if historical averages are used in the PEVs, (ii) y takes values 0, 1 or 2, which means that the agent buys its unpurchased tickets when the y^{th} hotel auction closes (0 means it does not wait at all), or the value S, which means that the version based on strategic demand reduction is used and (iii) z characterizes the aggressiveness with which the agent bids for hotel rooms and takes values L,M and H for low, medium and high degree of aggressiveness respectively.

To formally evaluate whether one version outperforms another, we use *paired t-tests*; values of less than 10% are considered to indicate a statistically significant difference (in most experiments the values are actually well below 5%). If more than 1 instances of a certain version participate in an experiment, we compute the t-test for all possible combinations of instances.[7]

The first set of experiments were aimed at verifying our observation that modeling the plane ticket prices improves the performance of the agent. We expected an improvement[8], since the agent uses this information to bid later for tickets whose price will not increase much (therefore achieving a greater flexibility at low cost), while bidding earlier for tickets whose price increases faster (reducing its cost). We run 2 experiments with the following 4 versions: WB-N2L, WB-M2L, WB-M2M and WB-M2H. In the first we run 2 instances of each agent, while in the second we run only one and the other 4 slots were filled with the standard agent provided by the TAC support team. The result are presented in table 1. The other agents, which model the plane ticket prices, perform better than agent WB-N2L, which does not do so. The differences between WB-N2L and the other agents are statistically significant, except for the one between WB-N2L and WB-M2L in experiment 2. We also observe that WB-M2L is outperformed by agents WB-M2M and WB-M2H, which in turn achieve similar scores; these results are statistically significant for experiment 1. Having determined that this modeling leads to significant improvement, we concentrated our attention only to agents using this feature.

The next experiment was designed to explore the tradeoff of bid aggressiveness. As proposed by our methodology we used agents WB-M2z (z=L,M,H), keeping all other partial strategies fixed, and we used a constant number of 2 instances of agent WB-M2M, while the number of agents WB-M2H was increased from 0 to 6. The rest of the slots were filled with instances of version WB-M2L. The result of this experiment are presented in table 2. By increasing the number of agents which bid more aggressively, there is more competition between agents and the hotel room prices increase, leading to a decrease in scores. While the number of aggressive agents #WB-M2H$\leq$4, the decrease in score is relatively small for all agents and is approximately linear with #WB-M2H; The aggressive agents (WB-M2H) do relatively better in less competitive environments and non-aggressive agents (WB-M2L) do relatively better in more competitive environments, but still not good enough compared to WB-M2M and WB-M2H agents. Overall WB-M2M (medium aggressiveness) performs comparably or better than the other agents in almost every instance. Agents WB-M2L are at a disadvantage compared to the other agents because they do not bid aggressively enough to acquire the hotel rooms that they need. When an agent fails to get a hotel room it needs, its score suffers a double penalty: (i) it will have to buy at least one more plane ticket at a high price in order to complete the itinerary, or else it will end up wasting at least some of the other commodities it has already bought for that itinerary and (ii) since the arrival and/or departure date will probably be further away from the customer's preference and the stay will be shorter (hence less entertainment tickets can be assigned), there is a significant utility penalty for the new itinerary. On the other hand, aggressive agents (WB-M2H) will not face this problem and they will score well in the case that prices do not go up. In the case that there are a lot of them in the game though, the price wars will hurt them more than other agents. The reasons for this are: (i) aggressive agents will pay more than other agents, since the prices will rise faster for the rooms that they need the most in comparison to other rooms, which are needed mostly by less aggressive agents, and (ii) the utility penalty for losing a hotel room becomes comparable to the price paid for buying the room, so non-aggressive agents suffer only a relatively small penalty for being overbid. Agent WB-M2M performs "reasonably well" in every situation, since it bids enough to maximize the probability that it is not outbid for critical rooms, and avoids "price wars" to a larger degree then WB-M2H. Based on these results we did not use low aggressiveness agents in next experiments.

The next set of experiments intended to further explore the tradeoff of bidding early for plane tickets against waiting more in order to gain more flexibility in planning. Initially we run a smaller experiment with 2 instances of each of the following agents: WB-M2M and WB-M2H together with WB-M1M (which bids for most of its tickets at the beginning) and WB-M0H (which buys immediately all

[6]WB-xyz is the based on our 2001 TAC agent and WB*xyz is a slightly improved version based on our 2002 TAC agent.

[7]This means that 8 t-tests will be computed if we have 2 instances of version A and 4 of version B etc. We consider the difference between the scores of A and B to be significant, if almost all the tests produce values below 10%.

[8]A rough estimate before the experiment was a gain of 120 to 150.

	Agent Scores								Average Scores			Statistically Significant Difference?		
#WB-M2H	1	2	3	4	5	6	7	8	WB-M2L	WB-M2M	WB-M2H	M2L/M2M	M2M/M2H	M2L/M2H
0 (178)	2614	2638	2490	2463	2421	2442	2526	2455	2466	2626	N/A	✓		
2 (242)	2339	2350	2269	2265	2229	2241	2347	2371	2251	2344	2359	✓	×	✓
4 (199)	2130	2073	2072	2029	2046	2098	2048	2033	2051	2101	2056	×	×	×
6 (100)	1112	1165	796	843	920	884	848	898	N/A	1138	865		✓	

Table 2: Scores for agents WB-M2L, WB-M2M and WB-M2H as the number of aggressive agents (WB-M2H) participating increases. In each experiment agents 1 and 2 are instances of WB-M2M. The agents above the stair-step line are WB-M2L, while the ones below are WB-M2H. The averages scores for each agent type are presented in the next rows. In the last rows, ✓indicates statistically significant difference in the scores of the corresponding agents, while × indicates statistically similar scores.

	Agent Scores								Average Scores		Statistically Significant Difference?
#WB-M0H	1	2	3	4	5	6	7	8	WB-M2M	WB-M0H	WB-M2M/WB-M0H
2 (343)	1607	1522	1564	1523	1497	1517	1531	1501	1517	1540	×
4 (282)	1398	1425	1401	1387	1292	1333	1265	1341	1403	1308	×
6 (69)	1570	1602	1278	1178	1241	1289	1151	1207	1586	1224	✓

Table 3: Scores for agents WB-M2M and WB-M0H as the number of early bidding agents (WB-M0H) participating increases. The agents above the stair-step line are WB-M2M, while the ones below are WB-M0H.

the plane tickets it needs and bids aggressively for hotel rooms.[9] We ran 78 games and observed that WB-M2M scores slightly higher than the other agents, while WB-M2H scores slightly lower. These results are however not statistically significant. The bigger experiment was done to examine the behavior of the two boundary strategies against each other. We varied the mixture of agents WB-M2M and WB-M0H as shown in table 3. When only 2 of the agents were WB-M0H, the WB-M0H's scored on average close to the score of the WB-M2M's, but as their number increased their score dropped and, when they are the majority, the WB-M2M's performed much worse than the WB-M2M's. In this case, the WB-M2M's try to stay clear of rooms whose price increases too much (usually, but not always, successfully), while the early-bidders do not have this choice due to the reduced flexibility in changing their plans. One interesting result which we did not expect is that the score of the WB-M2M's increases when there are 2 instances of them compared to the case when there are 4; however hotel room prices are higher in the former case, so this result seems contradictory! The explanation for this is that the prices tend to increase quite fast for the rooms that are needed by the early-bidders, so the 2 WB-M2M's avoid these rooms when possible and try to position themselves mostly on the other rooms, so they do not have to pay so much. This behavior also happens in the case that there are 4 WB-M2M's, but in this case there are many WB-M2M's and when they try to move away from the rooms that the early-bidders want, they end up on similar rooms (so the reason it's harder to find the good deals is because they stop being "deals" much more often once the other WB-M2M's go after them).

These results would normally allow us to conclude that it is usually beneficial not to bid for everything at the beginning of the game, but there is a minor catch: without using historical prices the early bidder agent buys goods "blindly". Therefore we introduced this feature and run an experiment in which we examined the benefit that agents WB*M2M, WB*M2H, WB*MSM and WB*MSH gain if historical prices are used. Note that we did not use WB*M0z, because the agent WB*MSz also buys the vast majority of its tickets at the beginning (but not all). The results are presented in table 4. We observe that the agents which bid earlier are the ones who benefit from the use of this feature, while the benefits for WB*M2M and WB*M2H are virtually non-existent. The increase of the price estimate has the effect that the planner generates itineraries which use slightly fewer rooms than before. This decreases the price wars between agents and improves their scores.

[9]An early-bidder must be aggressive, because if it fails to get a room, it will pay a substantial cost for changing its plan, due to the lack of flexibility in planning.

The last experiment extends the experiment presented in table 3. This time we examine the effect on agents WB*AyM and WB*AyH (the medium and high aggressiveness with historical prices in the PEVs at the beginning of the game) when $y = 0$, $y = 2$ and $y = S$. Since $y = S$ is the intermediate strategy we always keep 2 agents WB*ASz (z=M,H) in the mixture of agents and change the number of the other agents (which use the boundary strategies) as described by our methodology; half of these are of Medium and half of High aggressiveness. The results are presented in table 5. While in some cases more games are needed to make the results statistically significant, we observe that the strategy $y = 2$ which leaves the highest number of unpurchased tickets performs worse than the other two. The other two perform similarly overall. The only case, in which the WB*ASz's performance is statistically better than that of the early bidders, is when there are lots of early bidders. From these results we determine that the strategic demand agent is probably performing more consistently and that is the reason we used it in the TAC. Another observation that one might make is that the scores of all agents tend to go up as the prices go higher. We believe (but need to check further) that this is a result of the fact the historical prices are used in the PEVs mainly at the beginning of the game and when some auctions have closed we do not; the later bidding agents ($y = 2$) observe the lower prices and try to purchase more rooms which in turn drives the prices up. As their number decreases the economy becomes more efficient and all the agents profit.

We are continuing the experiments in order to increase the statistical confidence in the interpretation of the results so far. This is quite a time-consuming process, since each game is run at 15 to 20 minute intervals[10]. It took over to 4000 runs (about 8 weeks of continuous running time) to get the controlled experiment results and some 2000 more for our observations during the competitions.

5. TRADING AGENT COMPETITION: RESULTS AND OBSERVATIONS

We have entered our agent, WhiteBear, in the last two Trading Agent Competitions. Preliminary and seeding rounds were held before the finals, so that teams could improve their strategies over time. The top 16 teams were invited to participate in the semi-finals and the best 8 advanced to the finals.

In the 2001 TAC the *White Bear* variant we used in the competition was **WB-M2M**. The 4 top scoring agents (scores in parentheses) that year were: livingagents (3670), ATTac (3622), WhiteBear (3513) and Urlaub01 (3421). The scores in the finals were

[10]This is a restriction of the game and the TAC servers

#WB*A0z	Agent Scores								Average Scores				Statistically Significant Difference?		
	WB*ASM	WB*ASH	3	4	5	6	7	8	WB*A2M	WB*A2H	WB*A0M	WB*A0H	A2z/ASz	ASz/A0z	A2z/A0z
0 (413)	*2175*	2213	*1948*	*1954*	*1907*	1899	1939	1876	*1936*	1904	*N/A*	N/A	M✓ H✓		
2 (405)	*2148*	2174	*2053*	*2091*	1918	1995	*2159*	2234	*2072*	1957	*2159*	2234	M? H✓	M× H?	M? H✓
4 (438)	*2404*	2419	*2261*	2207	*2305*	*2390*	2408	2362	*2261*	2207	*2347*	2385	M✓ H✓	M× H×	M✓ H✓
6 (759)	*2495*	2528	*2459*	*2461*	*2443*	2442	2465	2462	*N/A*	N/A	*2454*	2456		M? H✓	

Table 5: Scores for agents WB*A2z, WB*ASz and WB*A0z (where z=M or H) as the number of early bidding agents (WB-A2z) participating increases. In each experiment agents 1 is WB*ASM and 2 is WB*ASH. The agents above the stair-step line are WB*A2z, while the ones below are WB*A0z (the scores when z=M are presented in *italic* and they are the first half). In the last rows, we compare the scores for the M and H aggressiveness of the two version; so M✓ in the A2z/Asz box means that the difference between A2M and ASM is significant. The ? means that with more games the difference could become statistically significant.

higher than in the previous rounds, because most teams had learned (ourselves included) that it was generally better to have a more adaptive agent than to bid too aggressively.[11] A surprising exception to this rule was livingagents [3] which followed a strategy similar to WB-A0H (it used historical prices to approximate closing prices). This agent capitalized on the fact that the other agents were careful not to be very aggressive and that prices remained quite low. Despite bidding aggressively for rooms, since prices did not go up, it was not penalized for this behavior. This strategy works well within the confines of an "efficient economy". This was observed in the last experiment that we ran as well. The other top scorer, ATTac, also purchased over 75% of the plane tickets at the beginning of the game. The fact that they deferred few of their plane ticket purchases for a later time is likely the main reasons for their success in 2001.

In the recent (2002) TAC, we implemented all the features and continued our controlled experiments. Using the knowledge obtain from the previous competition, we decided to enter version **WB*ASM** in the finals. This decision was also based on observations of the other agents' overall behavior. The 4 top scoring agents were: WhiteBear (3556), SouthamptonTAC (3492), Thalis (3351) and UMBCTAC (3320). The competence of the agents who qualified for the finals was much higher than in the previous year. The agents would bid more aggressively and most of them would buy all of the plane tickets at the beginning. However this did not lead to an inefficient economy, as most agents were more adaptive and restricted their itineraries to somewhat fewer rooms.[12]

6. CONCLUSIONS

In this paper we have proposed an architecture for bidding strategically in simultaneous auctions. We have demonstrated how to maximize the flexibility (actions available, information etc.) of the agent while minimizing the cost that it has to pay for this benefit, and that by using simple knowledge (such as modeling prices) of the domain it can make this choice more intelligently and improve its performance even more. We also showed that bidding aggressively is not a panacea and established that an agent, who is just aggressive enough to implement its plan efficiently, outperforms overall agents who are either not aggressive enough or who are too aggressive. Finally we established that even though generating a good plan is crucial for the agent to maximize its utility, the greedy algorithm that we used was more than capable to help the agent produce comparable results with other agents that use a slower provably optimal algorithm. One of the primary benefits of our approach is that it allowed us to combine seamlessly both principled methods and methods based on empirical knowledge, which, we believe, led to the consistently good performance in the TAC competitions. Overall our agent is adaptive, scalable, and robust, and its elements are general enough to work well in a multiple simultaneous auctions setting.

In the future we will continue to run experiments in order to further determine parameters that affect the performance of agents in multi-agent systems. We also intend to incorporate learning into our agent to evaluate how much this improves performance.

7. ACKNOWLEDGEMENTS

We would like to thank the organizers of both competitions for their technical support during the running of our experiments and the competition (over 6000 games!).

8. REFERENCES

[1] P. Anthony, W. Hall, V. Dang, and N. R. Jennings. Autonomous agents for participating in multiple on-line auctions. In *Proc IJCAI Workshop on E-Business and Intelligent Web*, pages 54–64, 2001.

[2] R. Brooks. Intelligence without reason. In *Proc of the 12th International Joint Conference on Artificial Intelligence*, 1999.

[3] C. Fritcshi and K. Dorer. Agent oriented software engineering for successful tac participation. In *Proc of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2002.

[4] Y. Fujishima, K. Leyton-Brown, and Y. Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proc of the 16th International Joint Conference on Artificial Intelligence*, pages 548–553, Aug. 1999.

[5] A. Greenwald and P. Stone. Autonomous bidding agents in the trading agent competition. *IEEE Internet Computing, April*, 2001.

[6] A. R. Greenwald and J. Boyan. Bid determination for simultaneous auctions. In *Proc of the 3rd ACM Conference on Electronic Commerce (EC-01)*, pages 115–124 and 210–212, Oct. 2001.

[7] A. R. Greenwald, J. O. Kephart, and G. J. Tesauro. Strategic pricebot dynamics. In *Proceedings of the ACM Conference on Electronic Commerce (EC-99)*, pages 58–67, Nov. 1999.

[8] T. Ito, N. Fukuta, T. Shintani, and K. Sycara. Biddingbot: a multiagent support system for cooperative bidding in multiple auctions. In *Proceedings of the Fourth International Conference on MultiAgent Systems*, pages 399 – 400, July 2000.

[9] P. Klemperer. Auction theory: A guide to the literature. *Journal of Economic Surveys Vol13(3)*, July 1999.

[10] C. Preist, C. Bartolini, and I. Phillips. Algorithm design for agents which participate in multiple simultaneous auctions. *In Agent Mediated Electronic Commerce III (LNAI), Springer-Verlag, Berlin*, pages 139–154, 2001.

[11] T. Sandholm and S. Suri. Improved algorithms for optimal winner determination in combinatorial auctions. In *Proc 7th Conference on Artificial Intelligence (AAAI-00)*, pages 90–97, July 2000.

[12] P. Stone, M. L. Littman, S. Singh, and M. Kearns. ATTac-2000: an adaptive autonomous bidding agent. In *Proc 5th International Conference on Autonomous Agents*, pages 238–245, May 2001.

[13] P. Stone, R. Schapire, M. Littman, J. Csirik, and D. McAllester. Attac-2001: A learning, autonomous bidding agent. In *Agent Mediated Electronic Commerce IV. LNCS, volume 2531. Springer Verlag, Berlin.*, 2002.

[14] I. A. Vetsikas and B. Selman. Whitebear: An empirical study of design tradeoffs for autonomous trading agents. In *Proc AAAI Workshop on Game Theoretic and Decision Theoretic Agents*, 2002.

[15] R. Weber. Making more from less: Strategic demand reduction in the fcc spectrum auctions. *Journal of Economics and Management Strategy Vol6(3)*, pages 529–548, 1997.

[16] M. P. Wellman, P. R. Wurman, K. O'Malley, R. Bangera, S. de Lin, D. Reeves, and W. E. Walsh. Designing the market game for tac. *IEEE Internet Computing, April*, March/April 2001.

[11] This was demonstrated by the second controlled experiment that we ran as well.

[12] In the same way that the inclusion of historical average prices in the PEVs produced the same effect on our agent.